

Git bash

Repositorio local: es el que tiene todos los archivos (que hayas guardado en él) en nuestra computadora.

commits: son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor

comandos:

ls -R

ls -l (Muestra toda la información de una carpeta: usuario, grupo, permisos, tamaño, fecha y hora de creación.)

ls -t (comando para ver la lista de archivos y los ultimos modificados o trabajador)

cp (comando para copiar archivos)

rm (elimina un archivo)

rmdir (elimina una carpeta vacia)

rm -r (elimina una carpeta y su contenido)

more (muestra el contenido de un archivo)

ls -a (muestra todos los archivos ocultos)

mv (mueve archivos, renombra las carpetas (el nombre que le vamos a poner a nuestra carpeta no deberia existir))

mv (mueve el contenido de carpeta1 a carpeta2)

cp (copia un archivo o carpeta)

" " Nos permiten utilizar términos que consistan en más de una palabra

. Permite hacer referencia al directorio donde estamos ubicados actualmente

cd/name/name (Cambia a una carpeta específica)

git init (genera un repositorio local, el cual almacena archivos)

git status

git add "nombre del archivo"(para agregar los comandos al repositorio)

git add. (agrega todos los archivos automaticamente)

git commit -m (confirmacion de los archivos que estamos agregando)

git log (muestra todos los commits realizados)

git remote add origin + URL de nuestro repositorio (comando para conectar git con github)

git remote -v (para verificar si nuestra conexion a nuestro repositorio en github fue realizada con exito)

git push (se utiliza para enviar archivos a nuestro repositorio en la nube)

git push origin master (se utiliza para indicar que vamos a llevar nuestros archivos de nuestro repositorio local al repositorio remoto e ingresarlos en la rama principal que se llama master)

git clone + URLdelRepositorioRemoto (permite crear una copia exasta de todos los archivos que existen en el repositorio remoto) Este comando solo se ejecuta 1 sola vez

git pull (baja de la nube o actualiza los archivos que se encuentran en el repositorio o que hayan sufrido algun tipo de modificacion)

git pull origin main (para indicarle que queremos traer todo el contenido actualizado de la rama principal)

git branch "nombreDeLaRama" (comando para crear una nueva rama en git)

git branch (para ver las ramas creadas)

git branch -d "nombreDeLaRama"(Elimina la rama llamada <branch>. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.)

git branch -D (Fuerza la eliminacion de la rama especificada, incluso si tiene cambios sin fusionar)

git checkout nombre_rama (Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si

tenemos cambios, para cambiarnos de rama, debemos:

1. Eliminarlos (deshaciendo los cambios).
2. Confirmarlos (haciendo un git commit.).)

Guardar cambios y subirlos al repositorio remoto

Una vez que terminamos de realizar los cambios que queremos en nuestra branch, ejecutamos los mismos comandos que vimos hasta ahora: git add, git commit, git status y git log. Pero cuando queramos subir esos cambios, debemos utilizar git push con el nombre de la rama en que estamos posicionados:

- git push origin <branch>

Así también, para traer los cambios de esa rama utilizamos el git pull agregando desde donde queremos traer los cambios:

- git pull origin <branch>

git branch nombre de rama a crear

git checkout nombre de rama para movernos a esa rama

/ = raiz (rutas absoluta si lleva /)

(ruta relativa no lleva /)